

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC VÀ KỸ THUẬT MÁY TÍNH



MÔN HỌC: VI XỬ LÝ - VI ĐIỀU KHIỂN (TN)

LAB 1 - LED ANIMATIONS

Giảng viên hướng dẫn: Lê Trọng Nhân
Tôn Huỳnh Long

Lớp: L02

Sinh viên thực hiện: Võ Thị Ánh Tuyết - 2213826

Thành phố Hồ Chí Minh, Tháng 09 Năm 2025



Contents

1	Exercise 1	2
2	Exercise 2	3
3	Exercise 3	4
4	Exercise 4	6
5	Exercise 5	8
6	Exercise 6	12
7	Exercise 7	13
8	Exercise 8	13
9	Exercise 9	13
10	Exercise 10	14
11	Link	14

1 Exercise 1

Report 1: Depict the schematic from Proteus simulation in this report. The caption of the figure

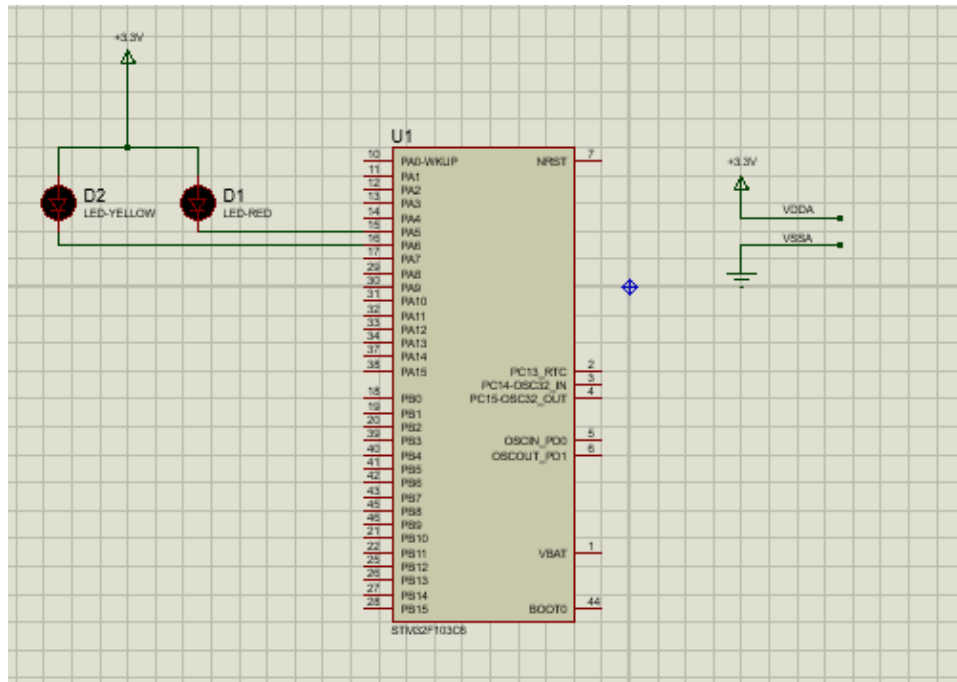


Figure 1: [Link to the figure](#)

Report 2: Present the source code in the infinite loop while of your project. If a userdefined functions is used, it is required to present in this part. A brief description can be added for this function (e.g. using comments). A template to present your source code is presented bellow.

```
1 int cnt = 0;
2 while (1)
3 {
4     switch (cnt){
5     case 4:
6         cnt = 0;
7     case 0:
8         HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin, RESET);
9         HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port, LED_YELLOW_Pin, SET
10    );
11         break;
```

```

11     case 2:
12         HAL_GPIO_WritePin(LED_RED_GPIO_Port, LED_RED_Pin, SET);
13         HAL_GPIO_WritePin(LED_YELLOW_GPIO_Port, LED_YELLOW_Pin,
14         RESET);
15         break;
16     default:
17         break;
18     }
19     cnt++;
20     HAL_Delay(1000);
21 }

```

Listing 1: Source code of while function

2 Exercise 2

Report 1: Present the schematic.

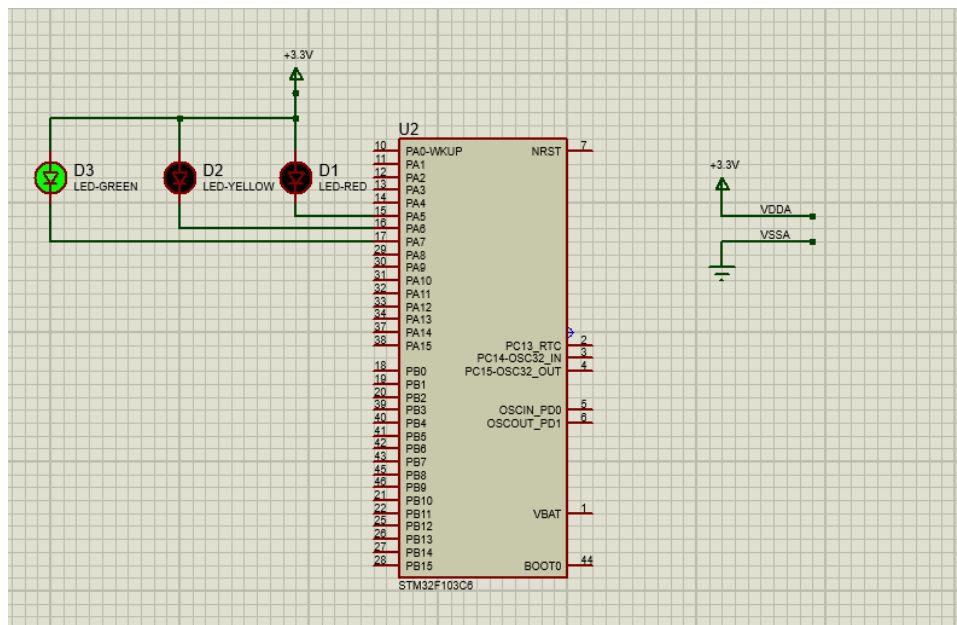


Figure 2: [Link to the figure](#)

Report 2: Present the source code in while.

```

1 void set_Leds(GPIO_PinState redstate, GPIO_PinState yellowstate,
2   GPIO_PinState greenstate){
3     HAL_GPIO_WritePin(GPIOA, GPIO_PIN_5, redstate);

```

```
3 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_6, yellowstate);  
4 HAL_GPIO_WritePin(GPIOA, GPIO_PIN_7, greenstate);  
5 }
```

Listing 2: Source code of set_LEDS function

```
1 int cnt = 0;  
2 while (1)  
3 {  
4     switch (cnt)  
5     {  
6         case 10:  
7             set_Leds(1, 1, 0);  
8             break;  
9         case 7:  
10            set_Leds(1, 0, 1);  
11            break;  
12         case 5:  
13            set_Leds(0, 1, 1);  
14            break;  
15         case 1:  
16            cnt = 11;  
17         default:  
18            break;  
19     }  
20     cnt--;  
21     HAL_Delay(1000);  
22 }
```

Listing 3: Source code of while function

3 Exercise 3

Extend to the 4-way traffic light. Arrange 12 LEDs in a nice shape to simulate the behaviors of a traffic light. A reference design can be found in the figure bellow

Note: The below schematic is used for Exercise 3, 4 and 5.

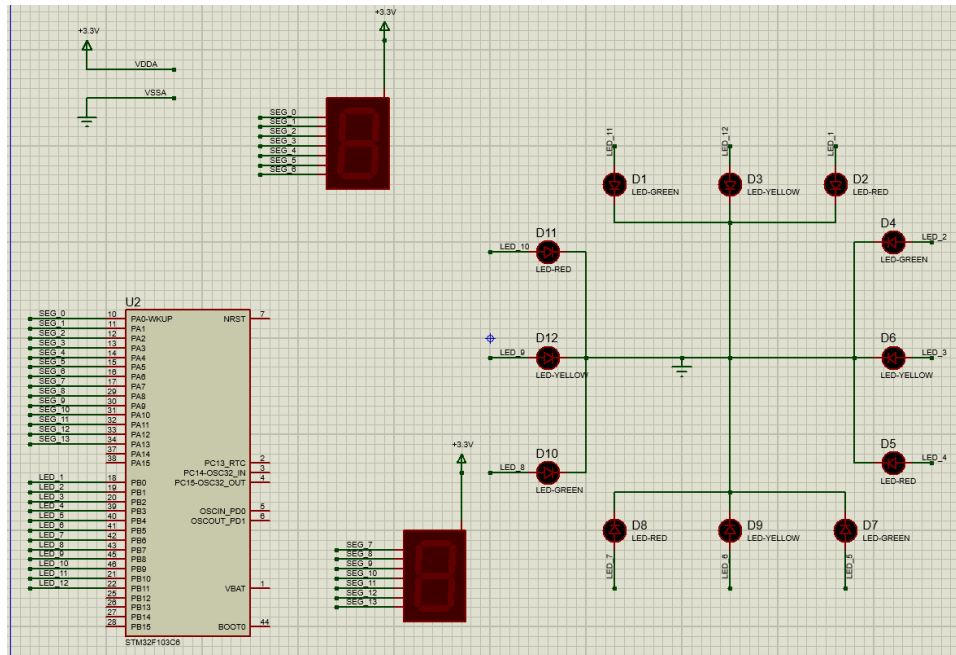


Figure 3: [Link to the figure](#)

```

1 GPIO_TypeDef *LED_PORTS[12] = {
2     LED_1_GPIO_Port, LED_2_GPIO_Port, LED_3_GPIO_Port,
3     LED_4_GPIO_Port, LED_5_GPIO_Port, LED_6_GPIO_Port,
4     LED_7_GPIO_Port, LED_8_GPIO_Port, LED_9_GPIO_Port,
5     LED_10_GPIO_Port, LED_11_GPIO_Port, LED_12_GPIO_Port
6 };
7
8 uint16_t LED_PINS[12] = {
9     LED_1_Pin, LED_2_Pin, LED_3_Pin,
10    LED_4_Pin, LED_5_Pin, LED_6_Pin,
11    LED_7_Pin, LED_8_Pin, LED_9_Pin,
12    LED_10_Pin, LED_11_Pin, LED_12_Pin
13 };
14
15 GPIO_PinState LEDS_state[4][12] = {
16     { 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0 },
17     { 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0 },
18     { 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0 },
19     { 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1 }
20 };

```

Listing 4: Source code of set_LEDS function

```
1 int cnt = 10;
2 while (1)
3 {
4     switch (cnt) {
5         case 10:
6             set_LEDS(LEDState[0]);
7             break;
8         case 7:
9             set_LEDS(LEDState[1]);
10            break;
11         case 5:
12             set_LEDS(LEDState[2]);
13            break;
14         case 2:
15             set_LEDS(LEDState[3]);
16            break;
17         case 1:
18             cnt = 11;
19             break;
20         default:
21             break;
22     }
23     cnt--;
24     HAL_Delay(1000);
25 }
```

Listing 5: Source code of while function

4 Exercise 4

Report 1: Present the schematic. *Checkout the schematic of Exercise3*

Report 2: Present the source code for display7SEG function.

```
1 GPIO_TypeDef *SEG_PORTS[12] = {
2     SEG_0_GPIO_Port, SEG_1_GPIO_Port, SEG_2_GPIO_Port,
3     SEG_3_GPIO_Port, SEG_4_GPIO_Port, SEG_5_GPIO_Port,
4     SEG_6_GPIO_Port
5 };
6
7 uint16_t SEG_PINS[12] = {
```

```
8     SEG_0_Pin, SEG_1_Pin, SEG_2_Pin, SEG_3_Pin,
9     SEG_4_Pin, SEG_5_Pin, SEG_6_Pin
10 };
11
12 GPIO_PinState LEDS_state[10][7] = {
13     { 0, 0, 0, 0, 0, 0, 1 },
14     { 1, 0, 0, 1, 1, 1, 1 },
15     { 0, 0, 1, 0, 0, 1, 0 },
16     { 0, 0, 0, 0, 1, 1, 0 },
17     { 1, 0, 0, 1, 1, 0, 0 },
18     { 0, 1, 0, 0, 1, 0, 0 },
19     { 0, 1, 0, 0, 0, 0, 0 },
20     {0, 0, 0, 1, 1, 1, 1 },
21     { 0, 0, 0, 0, 0, 0, 0 },
22     { 0, 0, 0, 0, 1, 0,0 }
23 };
24
25 void set_LEDS(GPIO_PinState *L_LEDS_state) {
26     for (int i = 0; i < 7; i++) {
27         HAL_GPIO_WritePin(SEG_PORTS[i], SEG_PINS[i], L_LEDS_state[i])
28         ;
29     }
30 }
31
32 void display7SEG(int number) {
33     switch (number) {
34         case 0:
35             set_LEDS(LEDS_state[0]);
36             break;
37         case 1:
38             set_LEDS(LEDS_state[1]);
39             break;
40         case 2:
41             set_LEDS(LEDS_state[2]);
42             break;
43         case 3:
44             set_LEDS(LEDS_state[3]);
45             break;
46         case 4:
```



```
46     set_LEDS(LED_state[4]);  
47     break;  
48     case 5:  
49         set_LEDS(LED_state[5]);  
50         break;  
51     case 6:  
52         set_LEDS(LED_state[6]);  
53         break;  
54     case 7:  
55         set_LEDS(LED_state[7]);  
56         break;  
57     case 8:  
58         set_LEDS(LED_state[8]);  
59         break;  
60     case 9:  
61         set_LEDS(LED_state[9]);  
62         break;  
63     default:  
64         break;  
65     }  
66 }
```

Listing 6: Source code of set_LEDS function and display7SEG function

```
1 int counter = 0;  
2 while (1)  
3 {  
4     if (counter >= 10)  
5         counter = 0;  
6     display7SEG(counter++);  
7     HAL_Delay(1000);  
8 }
```

Listing 7: Source code of while function

5 Exercise 5

```
1 GPIO_TypeDef *LED_PORTS[12] = {  
2     LED_1_GPIO_Port, LED_2_GPIO_Port, LED_3_GPIO_Port,  
3     LED_4_GPIO_Port, LED_5_GPIO_Port, LED_6_GPIO_Port,
```

```
4     LED_7_GPIO_Port, LED_8_GPIO_Port, LED_9_GPIO_Port,
5     LED_10_GPIO_Port, LED_11_GPIO_Port, LED_12_GPIO_Port
6 };
7
8 uint16_t LED_PINS[12] = {
9     LED_1_Pin, LED_2_Pin, LED_3_Pin,
10    LED_4_Pin, LED_5_Pin, LED_6_Pin,
11    LED_7_Pin, LED_8_Pin, LED_9_Pin,
12    LED_10_Pin, LED_11_Pin, LED_12_Pin
13 };
14
15 GPIO_PinState LEDS_state[4][12] = {
16     { 1, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0 },
17     { 1, 0, 1, 0, 0, 0, 1, 0, 1, 0, 0, 0 },
18     { 0, 0, 0, 1, 1, 0, 0, 0, 0, 1, 1, 0 },
19     { 0, 0, 0, 1, 0, 1, 0, 0, 0, 1, 0, 1 }
20 };
21
22 GPIO_TypeDef *SEG_PORTS[14] = {
23     SEG_0_GPIO_Port, SEG_1_GPIO_Port, SEG_2_GPIO_Port,
24     SEG_3_GPIO_Port, SEG_4_GPIO_Port, SEG_5_GPIO_Port,
25     SEG_6_GPIO_Port, SEG_7_GPIO_Port, SEG_8_GPIO_Port,
26     SEG_9_GPIO_Port, SEG_10_GPIO_Port, SEG_11_GPIO_Port,
27     SEG_12_GPIO_Port, SEG_13_GPIO_Port
28 };
29
30 uint16_t SEG_PINS[14] = {
31     SEG_0_Pin, SEG_1_Pin, SEG_2_Pin,
32     SEG_3_Pin, SEG_4_Pin, SEG_5_Pin,
33     SEG_6_Pin, SEG_7_Pin, SEG_8_Pin,
34     SEG_9_Pin, SEG_10_Pin, SEG_11_Pin,
35     SEG_12_Pin, SEG_13_Pin
36 };
37
38 GPIO_PinState LEDS_7SEG_state[6][7] = {
39     { 0, 0, 0, 0, 0, 0, 1 },
40     { 1, 0, 0, 1, 1, 1, 1 },
41     { 0, 0, 1, 0, 0, 1, 0 },
42     { 0, 0, 0, 0, 1, 1, 0 },
```

```
43     { 1, 0, 0, 1, 1, 0, 0 },
44     { 0, 1, 0, 0, 1, 0, 0 }
45 };
46 void set_LEDS(const GPIO_PinState *state){
47     for (int i = 0; i < 12; i++){
48         HAL_GPIO_WritePin(LED_PORTS[i], LED_PINS[i], state[i]);
49     }
50 }
51
52 void set_7SEG_X(const GPIO_PinState *L_LEDS_X_state){
53     for (int i = 0; i < 7; i++){
54         HAL_GPIO_WritePin(SEG_PORTS[i], SEG_PINS[i], L_LEDS_X_state[i
55         ]]);
56     }
57 }
58 void set_7SEG_Y(const GPIO_PinState *L_LEDS_Y_state){
59     for (int i = 0; i < 7; i++){
60         HAL_GPIO_WritePin(SEG_PORTS[i + 7], SEG_PINS[i + 7],
61         L_LEDS_Y_state[i]);
62     }
63 }
```

Listing 8: Source code of set_LEDS, set_7SEG_X and set_7SEG_Y function

```
1  while (1)
2  {
3      switch (cnt) {
4          case 10:
5              set_LEDS(LED_STATE[0]);
6              set_7SEG_Y(LED_7SEG_STATE[3]);
7              set_7SEG_X(LED_7SEG_STATE[5]);
8              break;
9          case 9:
10             set_7SEG_Y(LED_7SEG_STATE[2]);
11             set_7SEG_X(LED_7SEG_STATE[4]);
12             break;
13          case 8:
14             set_7SEG_Y(LED_7SEG_STATE[1]);
15             set_7SEG_X(LED_7SEG_STATE[3]);
```

```
16     break;
17 case 7:
18     set_LEDS(LEDState[1]);
19     set_7SEG_Y(LED7SEGState[2]);
20     set_7SEG_X(LED7SEGState[2]);
21     break;
22 case 6:
23     set_7SEG_Y(LED7SEGState[1]);
24     set_7SEG_X(LED7SEGState[1]);
25     break;
26 case 5:
27     set_LEDS(LEDState[2]);
28     set_7SEG_Y(LED7SEGState[5]);
29     set_7SEG_X(LED7SEGState[3]);
30     break;
31 case 4:
32     set_7SEG_Y(LED7SEGState[4]);
33     set_7SEG_X(LED7SEGState[2]);
34     break;
35 case 3:
36     set_7SEG_Y(LED7SEGState[3]);
37     set_7SEG_X(LED7SEGState[1]);
38     break;
39 case 2:
40     set_LEDS(LEDState[3]);
41     set_7SEG_Y(LED7SEGState[2]);
42     set_7SEG_X(LED7SEGState[2]);
43     break;
44 case 1:
45     set_7SEG_Y(LED7SEGState[1]);
46     set_7SEG_X(LED7SEGState[1]);
47     cnt = 11;
48     break;
49 default:
50     break;
51 }
52 cnt--;
53 HAL_Delay(1000); }
```

Listing 9: Source code of while function

6 Exercise 6

Report 1: Present the schematic.

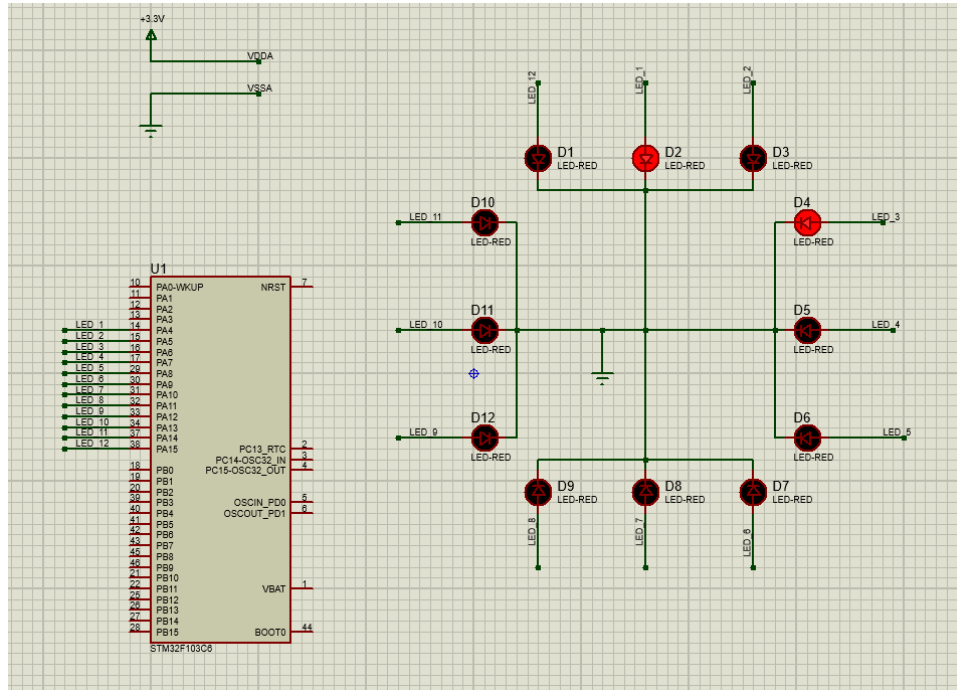


Figure 4: [Link to the figure](#)

Report 2: Implement a simple program to test the connection of every single LED. This testing program should turn every LED in a sequence.

```
1 GPIO_TypeDef *LED_PORTS[12] = {
2     LED_1_GPIO_Port, LED_2_GPIO_Port, LED_3_GPIO_Port,
3     LED_4_GPIO_Port, LED_5_GPIO_Port, LED_6_GPIO_Port,
4     LED_7_GPIO_Port, LED_8_GPIO_Port, LED_9_GPIO_Port,
5     LED_10_GPIO_Port, LED_11_GPIO_Port, LED_12_GPIO_Port
6 };
7
8 uint16_t LED_PINS[12] = {
9     LED_1_Pin, LED_2_Pin, LED_3_Pin,
10    LED_4_Pin, LED_5_Pin, LED_6_Pin,
11    LED_7_Pin, LED_8_Pin, LED_9_Pin,
12    LED_10_Pin, LED_11_Pin, LED_12_Pin
13 };
```

Listing 10: Source code of variables declaration

```
1 int cnt = 0;
2 while (1)
3 {
4     if (cnt >= 12) {
5         cnt = 0;
6     }
7     HAL_GPIO_WritePin(LED_PORTS[cnt], LED_PINS[cnt], 1);
8     HAL_Delay(1000);
9     HAL_GPIO_WritePin(LED_PORTS[cnt], LED_PINS[cnt], 0);
10    cnt++;
11 }
12
```

Listing 11: Source code of while function

7 Exercise 7

```
1 void clearAllClock() {
2     for (int i = 0; i < 12; i++) {
3         HAL_GPIO_WritePin(LED_PORTS[i], LED_PINS[i], 0); }}

```

Listing 12: Source code of clearAllClock function

8 Exercise 8

```
1 void setNumberOnClock(int num) {
2     HAL_GPIO_WritePin(LED_PORTS[num], LED_PINS[num], 1); }
3

```

Listing 13: Source code of setNumberOnClock function

9 Exercise 9

```
1 void clearNumberOnClock(int num) {
2     HAL_GPIO_WritePin(LED_PORTS[num], LED_PINS[num], 0); }
3

```

Listing 14: ource code of while function

10 Exercise 10

```
1 int cnt_hour = 0;
2 int cnt_min = 0;
3 int cnt_sec = 0;
4 while (1) {
5     setNumberOnClock(cnt_hour);
6     setNumberOnClock(cnt_min / 5);
7     setNumberOnClock(cnt_sec / 5);
8     HAL_Delay(1000);
9     cnt_sec++;
10    if (cnt_sec == 60) {
11        cnt_sec = 0;
12        cnt_min++;
13    }
14    if (cnt_min == 60) {
15        cnt_min = 0;
16        cnt_hour++;
17    }
18    if (cnt_hour == 12) {
19        cnt_hour = 0;
20    }
21    clearAllClock();
22 }
```

Listing 15: Source code of clearNumberOnClock

11 Link

[Link Demo](#)

[Link Github](#)